In the method of the invention for mapping the persistable data contained in the classes 92-96 onto a relational database, for each class 92-96, a corresponding peer class (peer classes 92a, 94a, 96a) is formed, and the persistable data in each of the classes 92-96 is passed to its corresponding peer class. The peer classes 92a-96a in turn map the persistable data onto a relational database to be stored in as persistent data.

The peer classes 92a -- 96a form an inheritance hierarchy. There is only one reference between the classes 92 -- 96 and their corresponding peer classes 92a -- 96a, namely the pointer iPeer in the root object (Abstract 1). The iPeer value is overwritten as classes are constructed down the inheritance tree (from top to bottom). Attributes stored in intermediate classes are still accessible from all the left hand column objects, since the (bottom right hand) object pointed to by the iPeer will inherit the attributes of all the classes above it in the right hand column. This advantageously saves a great deal of complexity in the code by obviating the need for every class on the left to have its own pointer to a corresponding class on the right. When an object on the right is retrieved from a database, code in "PersistablePeer1" can simply call "createOrigObject()", which will automatically call "createOrigObject" in the bottom right hand class, to automatically construct the correct object (& tree) in the left-hand column, matching the object retrieved.

Further understanding of the use of peer classes in the SAN management system of the invention can be obtained by reference to FIGURE X.

Administrator Notification

The SAN management system of the invention can notify the SAN operator/administrator of the occurrence of a condition, e.g., the utilization of a file system exceeding a threshold (e.g., defined by the host file system, the SAN administrator or otherwise). The SAN manager notifies the administrator of the first occurrence of the condition, but allows the administrator to define a time interval, herein referred to as alert interval, before the administrator is notified of subsequent occurrences of the same condition.

For example, the SAN management system may be monitoring a condition every 15 minutes, but the administrator may require a notification every two days. When the system detects an occurrence of the condition, it will determine whether it is the first time that the condition has been detected by consulting a database for date and time of a previous notification, if any, of the occurrence of the same condition. If there is no saved date and time corresponding to a previous notification, the manager transmits a notification to the SAN administrator, and saves the date and time of the transmittal. Alternatively, if the database contains a date and time corresponding to a previous notification of the same condition, the manager determines whether the time elapsed since the previous notification exceeds the alert interval. If the elapsed time exceeds the alter interval, a notification is transmitted. Otherwise, no notification is transmitted.

The use of an alert interval by the SAN management system of the invention allows an administrator to control the frequency of notifications sent by the manager thereto regarding the occurrences of various conditions. Further, the SAN management system preferably provides a

graphical user interface to the administrator for efficient and convenient setting of the alert interval.

Graphical User Interface

The SAN manager console employs a variety of graphical user interfaces (GUI) for displaying various components of the SAN, such as, the hosts, the storage devices, and their selected attributes to the SAN operator/administrator. As shown in FIGURE 15, a GUI server 98 communicates with the SAN Manager by utilizing, for example, an Object Request Broker (ORB) over a TCP/IP connection. The Manager can create objects (services) and "bind" them to the ORB directory service. GUI can "look up" an object by name in the directory service and get the object "proxy". GUI can invoke object methods to obtain information or to perform operations.

As an example of a GUI utilized by the SAN manager of the invention, FIGURE 16 illustrates a display 100 in a portion of which a storage device, and its selected attributes, such as, its serial number, its product Id, are shown. The display is presented on consoles or other graphical HMI devices of the type discussed above in connection with FIGURE 2. The Storage device is identified in a first panel, and its selected attributes are displayed in a second panel vertically separated from the first panel. In this illustrated embodiment, the selection of the storage device in the first panel, for example, by clicking on the icon representing the storage device, results in the display of its properties in the second panel.